# STFAR: Test-Time Adaptive Object Detection through Self-Training and Feature Alignment Regularization

Nanqing Liu[a,1], Yijin Chen[b], Yongyi Su[b,c], Haojie Zhang[b], Lile Cai[c], Heng-Chao Li[d], Kui Jia[e], Tianrui Li[f], Xun Xu[c,*], Chuan-Sheng Foo[c,g]

[a]*School of Information Science and Technology, Yunnan Normal University, Kunming, China*
[b]*School of Electrical and Information Engineering, South China University of Technology, Guangzhou, China*
[c]*Institute for Infocomm Research, A*STAR, Singapore, Singapore*
[d]*School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China*
[e]*School of Data Science, The Chinese University of Hong Kong (Shenzhen), Shenzhen, China*
[f]*School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China*
[g]*Centre for Frontier AI Research, A*STAR, Singapore, Singapore*

**Abstract**

Domain adaptation aims to improve the robustness of object detection models under distribution shifts between source and target domains. Unlike traditional methods requiring full access to target-domain data during training, this work focuses on Test-Time Adaptive Object Detection (TTAOD), where model adaptation occurs dynamically at test time with unlabeled target samples, following the general paradigm of Test-Time Adaptation (TTA). First, we employ a self-training framework to generate pseudo-labeled objects using an exponential moving average model. Since self-training can produce incorrect pseudo-labels, we further align feature distributions at two output levels: image level and instance level, as regularization to support self-training and help the target domain learn the source-domain distribution. At the image level, Global Feature Alignment (GFA) mitigates domain shift by aligning the backbone feature distributions between the source and target domains. At the instance level, Prototype-guided Foreground Contrast (PFC) improves feature discriminability by using source-domain prototypes to guide contrastive learning, pulling features of the same category closer together while pushing apart those of different categories. To validate the proposed approach, we construct benchmarks on several object detection datasets, simulating realistic distribution shifts, and adapt existing TTA methods for comparison. Extensive experiments show that our method significantly enhances detection robustness and achieves strong performance under both standard TTA and continual TTA settings. The code is available at `https://github.com/Lans1ng/STFAR`.

*Keywords:* Object Detection, Test-Time Adaptation, Feature Alignment, Contrastive Learning

## 1. Introduction

Object detection is a fundamental computer vision task with broad applications in autonomous driving Mushtaq et al. (2025), remote sensing Rajendran et al. (2025); Hou et al. (2026); Xu et al. (2026), medical imaging Cai et al. (2025), and industrial inspection Fan et al. (2025); Yang et al. (2025). With the rapid progress of deep neural networks, object detectors have achieved remarkable performance on natural images. However, robustness remains a major challenge. Prior studies have shown that photorealistic corruptions can cause substantial performance degradation in modern detectors.

To improve the robustness of object detection models, unsupervised domain adaptation approaches seek to learn domain-invariant features that improve model generalization Li et al. (2025); Zhao and Wang (2025). Recent studies have also explored domain-adaptive and robust object detection under challenging conditions Wen et al. (2025b,c,a). As shown in Fig. 1(a), UDA assumes both source- and target-domain samples are available when training a domain-generalizable model. This assumption, however, only applies to scenarios where source-domain data is accessible and the target-domain distribution is static. In more realistic scenarios, source-domain data may not be available for adaptation due to privacy constraints. Hence, jointly learning an invariant representation from both source and target data is not feasible. As shown in Fig. 1(b), source-free object detection (SFOD) Li et al. (2021, 2022) relaxes this requirement by adapting to the target domain without accessing source-domain data. Although SFOD moves closer to a realistic domain adaptation setup, we argue that important practical challenges remain unresolved. First, the target domain distribution, e.g., the types of corruption, is often unpredictable before testing begins. For example, under changing weather or lighting conditions on a new camera, the corruption type is of-

*Corresponding author: Xun Xu
*Email addresses:* `lansing163@163.com` (Nanqing Liu), `eechenyijin@mail.scut.edu.cn` (Yijin Chen), `eesuyongyi@mail.scut.edu.cn` (Yongyi Su), `eezhanghaojie@gmail.com` (Haojie Zhang), `cailile1988@gmail.com` (Lile Cai), `lihengchao_78@163.com` (Heng-Chao Li), `kuijia@cuhk.edu.cn` (Kui Jia), `trli@swjtu.edu.cn` (Tianrui Li), `alex.xun.xu@gmail.com` (Xun Xu), `foo_chuan_sheng@i2r.a-star.edu.sg` (Chuan-Sheng Foo)

[1]This work was partially conducted during internship at Centre for Frontier AI Research, A*STAR.

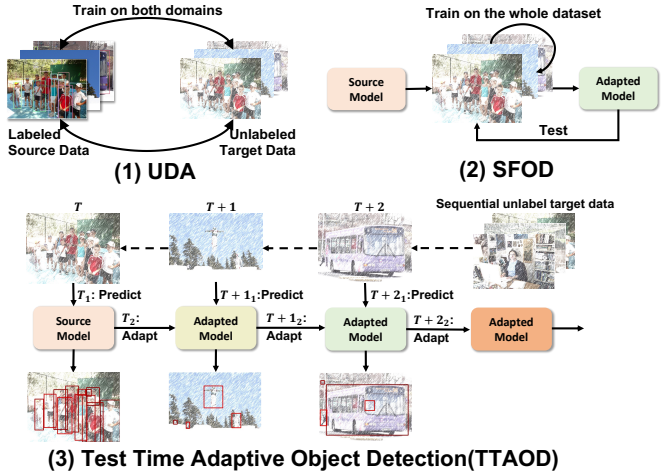


Figure 1: Overview of the adopted test-time adaptation protocol and its differences from UDA and SFOD. UDA requires access to both source- and target-domain data for adaptation. SFOD requires access to all target-domain test data for adaptation. In contrast, TTA sequentially adapts to target-domain test data on-the-fly. The dashed lines in (c) represent the sequential inference stream, indicating that test samples arrive one at a time and the model must simultaneously produce predictions and adapt its weights online.

ten unknown until test samples are observed. SFOD will thus struggle to adapt to a test distribution that is entirely unknown before testing starts. Moreover, test samples arrive sequentially, and predictions should be produced immediately upon the arrival of each new sample Su et al. (2022). Since SFOD requires access to the full target set for adaptation, it cannot support simultaneous inference and on-the-fly adaptation.

To address unpredictable target-domain distributions and the need for simultaneous inference and adaptation, test-time adaptation (TTA) Sun et al. (2020); Wang et al. (2020); Su et al. (2022) has emerged as a practical paradigm for updating model weights online. Under TTA, adaptation is performed sequentially at test time while predictions are produced immediately Su et al. (2022). Existing TTA methods typically rely on dynamic source-target distribution alignment Su et al. (2022), self-training with pseudo-labels, or auxiliary self-supervised tasks Sun et al. (2020). Recent continual TTA methods for classification, such as CoTTA Wang et al. (2022), RoTTA Yuan et al. (2023), and TCA Ni et al. (2025), further emphasize stability under dynamic domain shifts via parameter restoration, memory-based updates, or topology-preserving regularization. However, these approaches have been developed almost exclusively for image classification, and effective TTA for object detection remains underexplored. To fill this gap, we study test-time adaptive object detection (TTAOD). As illustrated in Fig. 1(c), TTAOD adopts a predict-then-adapt cycle: unlabeled test samples with potentially varying corruption types arrive sequentially (indicated by dashed arrows), and at each timestep the model first produces detection predictions on the incoming sample, then immediately updates its weights before processing the next one. This enables the model to continuously track distribution shifts on-the-fly, without any prior knowledge of the target domain distribution and without the need to collect target data in advance.

We study Test-Time Adaptive Object Detection (TTAOD) from two complementary perspectives. First, self-training provides a natural way to exploit unlabeled test samples: the detector predicts pseudo-labels and uses them as supervision for online adaptation. However, pseudo-labels in object detection can be noisy, and repeated adaptation to them can accumulate errors and induce confirmation bias Arazo et al. (2020). Self-training alone is therefore insufficiently stable for test-time adaptation. Distribution alignment has also shown promise in test-time adaptation Su et al. (2022). Building on this observation, we use alignment as a regularizer for self-training rather than as a stand-alone objective. At the image level, we introduce **Global Feature Alignment (GFA)** to align backbone feature distributions between the source and target domains, keeping the detector anchored to the source-domain representation during online adaptation. This helps preserve the effectiveness of the source-domain RPN and detection heads under corruption. At the instance level, object detection requires not only globally aligned features but also discriminative foreground representations for classification and localization. Yet target-domain predictions are often noisy, making direct instance-level alignment unreliable. We therefore introduce **Prototype-guided Foreground Contrast (PFC)**, which uses source-domain class prototypes to regularize target RoI features by pulling same-class features closer and pushing different-class features apart. Together, GFA and PFC provide image-level and instance-level regularization, making self-training more robust under test-time distribution shift.

To validate the effectiveness of the proposed method, we establish benchmarks for test-time adaptive object detection. We create corrupted target-domain benchmarks from three standard object detection datasets and adapt representative TTA methods to object detection. We further simulate a realistic continual domain-shift scenario Wang et al. (2022) by continuously adapting the model to multiple types of distribution shifts. Extensive experiments are carried out on these datasets. The contributions of this work are summarized below:

- We study the **Test-Time Adaptive Object Detection (TTAOD)** setting, where a detector adapts on-the-fly to unpredictable corruptions without source images at test time. We construct comprehensive benchmarks on COCO-C, PASCAL-C, Foggy Cityscapes, and Clipart, and adapt representative TTA methods to object detection as baselines, enabling systematic evaluation of this challenging setting.
- We propose **Global Feature Alignment (GFA)**, which aligns global backbone feature distributions between source and target domains using compact offline-stored first- and second-order statistics, without requiring source images at test time. GFA serves as a regularization term during self-training to prevent confirmation-bias-induced drift, making it well-suited to the online streaming inference protocol of TTA.
- We propose **Prototype-guided Foreground Contrast**

**(PFC)**, which enhances instance-level foreground feature discriminability by using source-domain class prototypes to guide contrastive learning, pulling same-class features closer and pushing different-class features apart to reduce category confusion from noisy pseudo-labels.

The remainder of this paper is organized as follows. Section 2 reviews related work on domain-adaptive object detection, source-free object detection, and test-time adaptation. Section 3 presents the proposed STFAR framework, including the self-training pipeline, Global Feature Alignment, and Prototype-guided Foreground Contrast. Section 4 reports extensive experiments and ablation studies under both single-domain and continual domain shifts. Section 5 concludes the paper.

## 2. Related Works

### *2.1. Domain Adaptive Object Detection*

In recent years, a number of unsupervised domain adaptation (UDA) methods have been proposed to reduce domain gaps in object detection. These methods can be grouped into three broad categories. i) Aligning source and target distributions at different layers and levels. For example, DA-Faster Chen et al. (2018), a pioneering UDA method, proposes a domain-adaptive Faster R-CNN that reduces discrepancy at both the image and instance levels by introducing two domain classifiers with adversarial training. SWDA Saito et al. (2019) aligns local features in shallow layers and image-level features in deep layers, i.e., strong local and weak global alignment. Similar to SWDA, HTCN Chen et al. (2020) and SSA-DA Zhao et al. (2020) align features on multiple layers by adversarial training. H2FA R-CNN Xu et al. (2022) studies cross-domain weakly supervised object detection and introduces holistic and hierarchical feature alignment. ii) Learning from noisy labels through self-training, e.g., NL Khodabandeh et al. (2019). iii) Sample-generation-based strategies that synthesize training data to bridge the domain gap. Recent related studies have also considered prompt-based feature alignment and robust detection under challenging visual conditions Wen et al. (2025b,c,a). In our method, a momentum-updated Faster R-CNN is used to improve stability during test-time adaptation. Although these methods achieve strong performance, they all require access to source-domain data during adaptation. When source data is inaccessible due to privacy constraints or storage overhead, more challenging settings arise, including source-free domain adaptation Liang et al. (2020); Li et al. (2022) and test-time adaptation Sun et al. (2020); Wang et al. (2020).

### *2.2. Source-Free Object Detection*

Without access to source data, Source-Free Domain Adaptation aims to adapt a source-pretrained model using only unlabeled target data. In classification, SHOT Liang et al. (2020) generates pseudo-labels for each target sample and performs a self-training process and information maximization to ensure class balance. Recently, a few source-free domain adaptation methods have been used to alleviate the domain gap in object detection tasks when source data is not accessible, a setting often referred to as Source-Free Object Detection (SFOD). HCL Huang et al. (2021) proposes historical contrastive instance discrimination to encourage the consistency between current representation and historical representations. LODS Li et al. (2022) enhances the style of the target image via the style enhancement module and reduces the style degree difference between the original image and the enhanced one. SFOD methods have been shown to perform well on cross-domain object detection, even when compared with UDA methods Li et al. (2022). Nevertheless, SFOD typically requires multiple adaptation epochs on the target domain. This makes it unsuitable for realistic deployment scenarios in which inference and adaptation must be performed simultaneously and target data cannot be collected in bulk beforehand.

### *2.3. Test-Time Adaptation*

Collecting target-domain samples in bulk in advance and adapting the source model offline limits deployment to static and known target domains. To enable fast online adaptation on unlabeled target data, Test-Time Adaptation (TTA) Sun et al. (2020); Wang et al. (2020); Fortes et al. (2026) has emerged. TTT-R Sun et al. (2020), a pioneer in this line, adapts the model on-the-fly by an auxiliary self-supervised task. Tent Wang et al. (2020) first proposes a fully test-time adaptation method without any auxiliary branch. Following TTT-R and Tent, many effective methods have been proposed, including source-target distribution alignment Su et al. (2022), self-training with pseudo-labels, prototype learning Iwasawa and Matsuo (2021), and more realistic test-time training protocols Su et al. (2022). However, existing TTA methods are designed almost exclusively for image classification rather than object detection, where on-the-fly adaptation is more challenging. In this work, we study the more practical problem of adapting an object detector to a real-time target domain on-the-fly, which we denote as Test-Time Adaptive Object Detection (TTAOD).

## 3. Methodology

### *3.1. Overview of Test-Time Adaptation*

Test-time adaptation aims to adapt model weights to the target-domain distribution alongside inference. We denote the source-domain labeled data as $D_s = \{x_i, y_i\}$, where $y_i = \{b_i, c_i\}$ denotes the ground-truth box annotations and class labels, respectively. We further denote the backbone features as $f_i = f(x_i; \Theta) \in \mathbb{R}^{H \times W \times C}$, the proposals after RPN and RoI pooling as $a_i \in \mathbb{R}^{N_a \times D}$, and the predictors as $h_c(a_i)$ for semantic classification and $h_r(a_i)$ for box regression. An object detection model is trained on the source-domain labeled data by optimizing the classification and regression losses. When the model is deployed on target-domain unlabeled data $D_t = \{x_j\}$, we assume that test samples arrive sequentially, predictions must be produced immediately, and the model is updated online using a small incoming mini-batch rather than a pre-collected

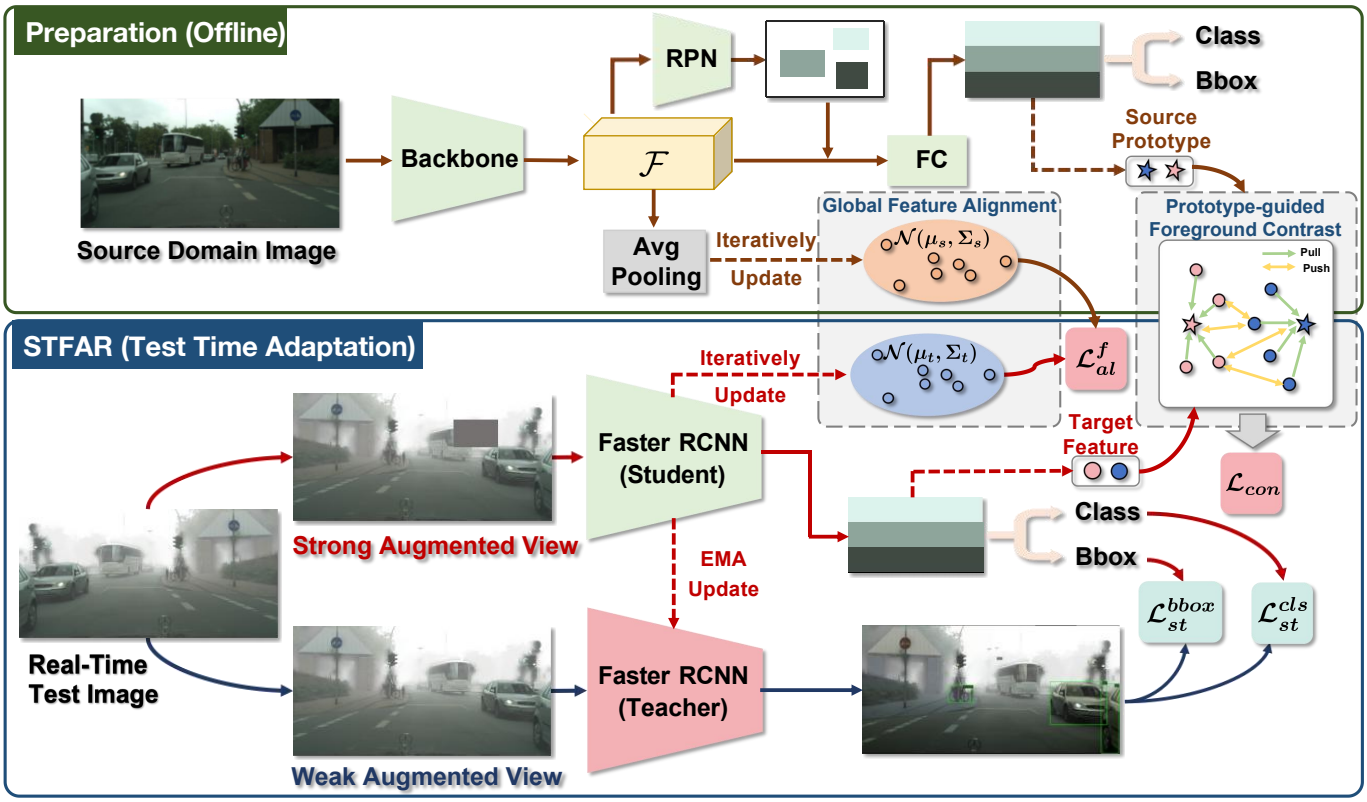


Figure 2: Overview of the proposed STFAR framework. In the source domain, STFAR computes feature distributions at both the global and foreground levels in an offline manner. During test-time adaptation, self-training is applied by predicting pseudo-labels with the teacher network. The student network is then supervised by the pseudo-labels. Self-training is further regularized by distribution alignment for improved robustness.

target set. In the following sections, we elaborate on how self-training with distribution alignment regularization enables this form of test-time adaptation for object detection.

### 3.2. Test-Time Adaptation by Self-Training

Self-training has proven highly effective in semi-supervised learning. It typically generates pseudo-labels for unlabeled samples, and the most confident predictions are then used to supervise model training. In this work, we adopt an approach similar to semi-supervised learning Tarvainen and Valpola (2017): two networks are maintained throughout adaptation, namely the student network $f(x;\Theta)$ and the teacher network $f(x;\hat{\Theta})$. The teacher-network weights are updated as the exponential moving average of the student weights, i.e., $\hat{\Theta}_t = \beta\hat{\Theta}_{t-1} + (1-\beta)\Theta_t$. Each input image is augmented with one strong augmentation $\mathcal{S}(x)$ and one weak augmentation $\mathcal{W}(x)$. We adopt the strong-augmentation strategy proposed in Xu et al. (2021). The teacher model predicts objects on the weakly augmented sample $\mathcal{W}(x)$ and obtains a set of pseudo-labeled objects $P = \{\hat{b}_i, \hat{y}_i\}$, where $\hat{b}_i$ and $\hat{y}_i$ denote the bounding-box coordinates and class labels, respectively. To reduce the impact of noisy pseudo-labels, only detections with prediction confidence above a threshold are retained before being passed to the student model. The student model then treats these pseudo-labeled objects as ground truth, and standard supervised losses are applied to the student branch. Specifically, we optimize the classification $\mathcal{L}_{st}^{cls}$ and regression $\mathcal{L}_{st}^{reg}$ losses on the student model branch, where the classification and regression losses follow the definitions in Ren et al. (2016).

### 3.3. Global Feature Alignment

Self-training alone is prone to the influence of incorrect pseudo-labels, a phenomenon known as confirmation bias Arazo et al. (2020). In semi-supervised learning, this issue is alleviated by the presence of labeled data, which provides strong regularization. However, in test-time adaptation where no labeled data is available, naive self-training may easily overfit to noisy predictions and lead to performance degradation. To improve the robustness of self-training, we incorporate distribution alignment as a regularization term, a strategy that has proven effective in prior TTA/TTT works Su et al. (2022). In this work, we propose **Global Feature Alignment (GFA)**, which aligns the distribution of global backbone features between the source and target domains. Unlike feature alignment methods in UDA (e.g., DA-Faster, SWDA) that require concurrent source images and treat alignment as the primary training objective, GFA is specifically designed for the TTA setting: (1) it requires no source images at test time, relying only on compact offline-stored first- and second-order source statistics; (2) it serves as a regularization term to stabilize self-training rather than as the primary objective; and (3) it operates online on the sequential test stream using small incoming mini-batches, compatible with streaming inference. This regularization keeps

backbone features closer to the source-domain distribution and helps suppress error accumulation during online adaptation.

Specifically, we model the global features in the source domain using a multivariate Gaussian distribution $N(\mu_s, \Sigma_s)$. In a typical Faster R-CNN pipeline, the backbone network outputs a global feature map $f_i \in \mathbb{R}^{C\times H\times W}$ for an input image $x_i$. To extract a compact representation for estimating the distribution, we perform average pooling over spatial dimensions:

$$g(x_i) = \frac{1}{HW}\sum_{h,w} f_{ihw} \tag{1}$$

With vectorized features for each image, we estimate the source-domain distribution parameters by Eq. 2.

$$\begin{aligned}\mu_s &= \frac{1}{|\mathcal{D}_s|}\sum_{x_i\in\mathcal{D}_s} g(x_i),\\ \Sigma_s &= \frac{1}{|\mathcal{D}_s|}\sum_{x_i\in\mathcal{D}_s} (g(x_i)-\mu_s)(g(x_i)-\mu_s)^\top\end{aligned} \tag{2}$$

We align the target distribution to the source domain by minimizing the symmetric KL divergence between two multivariate Gaussian distributions, as in Eq. 3. Since the KL divergence between two Gaussian distributions has a closed-form solution, the alignment objective can be optimized directly with gradient descent.

$$\mathcal{L}_{al} = D_{KL}(\mathcal{N}(\mu_s,\Sigma_s)\|\mathcal{N}(\mu_t,\Sigma_t)) + D_{KL}(\mathcal{N}(\mu_t,\Sigma_t)\|\mathcal{N}(\mu_s,\Sigma_s)) \tag{3}$$

Although the source-domain statistics can be computed offline using all available source training samples, estimating the target distribution under the TTA protocol is less straightforward. Estimating the target distribution from a single mini-batch is sensitive to the randomness of a small temporal window, because test samples are not necessarily drawn i.i.d. from the target-domain distribution Gong et al. (2022). Therefore, we estimate the target-domain distribution in an exponential moving average manner. Specifically, we use a hyperparameter $\gamma$ to control the contribution of the current mini-batch, and derive the target-domain distribution incrementally according to Eq. 4, where $B$ denotes a mini-batch of test samples.

$$\begin{aligned}\mu_t &= \mu_t + \delta, \quad \delta = \gamma\sum_{x_i\in\{B\}} (g(x_i)-\mu_t)\\ \Sigma_t &= \Sigma_t + \gamma\sum_{x_i\in\{B\}} [(g(x_i)-\mu_t)(g(x_i)-\mu_t)^\top - \Sigma_t] - \delta\delta^\top\end{aligned} \tag{4}$$

### *3.4. Prototype-guided Foreground Contrast*

Global feature alignment promotes consistent representation across entire images. However, unlike image classification, object detection relies heavily on accurate instance-level features. Existing methods, such as WHW Yoo et al. (2024), typically adopt distribution-alignment strategies to align instance-level features between the source and target domains. Yet, due to classification and regression errors, target-domain predictions are often noisy, leading to inaccurate feature distributions and category confusion. To address this issue, feature alignment alone is insufficient; cross-category contrastive learning is also needed to improve the discriminability of target-domain features. At the instance level, PFC uses source prototypes to make target RoI features more discriminative and reduce category confusion caused by noisy pseudo-labels.

To this end, inspired by FSCE Sun et al. (2021), a contrastive head is introduced alongside the classification and regression heads. This additional branch employs a two-layer multilayer perceptron (MLP) to project the RoI features $x \in \mathbb{R}^{N\times D}$ into a contrastive embedding $z \in \mathbb{R}^{N\times D_c}$, where $D$ is the RoI feature dimension and $D_c = 128$ is the default contrastive embedding dimension. The transformation is formulated as:

$$z = \mathbf{W}_2\cdot\sigma(\mathbf{W}_1\cdot x + b_1) + b_2, \tag{5}$$

where $\mathbf{W}_1, \mathbf{W}_2$ and $b_1, b_2$ denote the weights and biases of the two linear layers, and $\sigma(\cdot)$ represents the ReLU activation function.

We compute class prototypes offline from source-domain foreground RoI features. Specifically, for each class $c$, the source prototype is obtained by averaging all source-domain foreground RoI features assigned to that class:

$$\bar{a}_c = \frac{1}{N_c}\sum_{i=1}^{N_c} a_{ic}^s, \tag{6}$$

where $a_{ic}^s \in \mathbb{R}^D$ denotes the RoI feature of the $i$-th source-domain foreground instance from class $c$, and $N_c$ is the number of source-domain foreground instances belonging to class $c$. The source prototype in the contrastive embedding space is then computed by

$$z_c^s = \mathbf{W}_2\cdot\sigma(\mathbf{W}_1\cdot\bar{a}_c + b_1) + b_2. \tag{7}$$

During test-time adaptation, for each target RoI feature $z_i$ with pseudo-label $c_i$, we use the source prototype $z_{c_i}^s$ of the same class as the positive anchor, and the prototypes of all other classes as negatives. This yields a prototype-guided contrastive objective that pulls target foreground features toward the corresponding source-class prototype while separating them from prototypes of other classes. The contrastive loss is defined as:

$$\mathcal{L}_{con} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{z_i}, \tag{8}$$

$$\mathcal{L}_{z_i} = -\log\frac{\exp(\tilde{z}_i^\top\tilde{z}_{c_i}^s/\tau)}{\sum_{c'=1}^{C_n}\exp(\tilde{z}_i^\top\tilde{z}_{c'}^s/\tau)}, \tag{9}$$

where $C_n$ denotes the number of object categories, $\tau$ is the temperature hyperparameter in InfoNCE van den Oord et al. (2018), $\tilde{z}_i = \frac{z_i}{\|z_i\|} \in \mathbb{R}^{D_c}$ is the normalized target embedding, and $\tilde{z}_c^s = \frac{z_c^s}{\|z_c^s\|} \in \mathbb{R}^{D_c}$ is the normalized source prototype. In this way, PFC uses stable source-domain prototypes to regularize noisy target foreground features during online adaptation.

### 3.5. Overall TTA Algorithm for Object Detection

We summarize the overall test-time adaptation algorithm for object detection. In the source domain, we first extract backbone and foreground features offline. During test-time adaptation, the detector produces predictions for each incoming sample to support immediate inference while simultaneously updating the target-domain distribution estimates. Once a mini-batch of test samples has been accumulated, we update the model weights via gradient descent. A detailed description of the procedure is given in Alg. 1.

**Algorithm 1:** Test-time adaptive object detection algorithm

**Input:** Testing sample batch $\mathcal{B}^t = \{x_i\}_{i=1\cdots N_B}$.

# Offline Stage:
**for** $x_i \leftarrow 1$ **to** $N_B$ **do**
  Predict objects: $h_c(a(f(x_i))), h_r(a(f(x_i)))$;
# Adaptation Stage:
**for** $x_i \leftarrow 1$ **to** $N_B$ **do**
  Augmentation: $\mathcal{W}(x_i), \mathcal{S}(x_i)$;
  Pseudo label prediction: $\mathcal{P} = \{\hat{b}_{ij}, \hat{y}_{ij}\}$;
  Incremental distribution update: Update $\mathcal{N}(\mu_t, \Sigma_t)$ by Eq. 4;
  Test-time adaptation loss: $\mathcal{L}_{tta} = \lambda_{st}^{cls}\mathcal{L}_{st}^{cls} + \lambda_{st}^{reg}\mathcal{L}_{st}^{reg} + \lambda_{al}\mathcal{L}_{al} + \lambda_{con}\mathcal{L}_{con}$;
  Gradient descent update: $\Theta = \Theta - \alpha\nabla\mathcal{L}_{tta}$;

## 4. Experiments

### 4.1. Dataset and Evaluation Protocol

We evaluate STFAR on several standard object detection benchmarks. **MS-COCO**Lin et al. (2014) provides 118K labeled training images in train2017, 123K unlabeled images in unlabeled2017, and 5K validation images in val2017. We construct COCO-C by applying 15 common corruptions to the COCO validation set, and use the clean COCO training split to pretrain the source detector. **PASCAL**Everingham et al. (2015) contains 20 object categories. We use about 15K images from the 2007 and 2012 trainval splits for source training, and build PASCAL-C by corrupting the PASCAL VOC 2007 test set, which contains 4,952 images. **Cityscapes** Cordts et al. (2016) contains 2,975 training images and 500 validation images over 8 object categories collected under clear-weather conditions. **Rainy Cityscapes** Sindagi et al. (2020) synthesizes rain on Cityscapes images with a physics-based rendering pipeline. Following standard practice, we use the strongest rain setting ($\alpha$=0.03), which yields 99 test images. **Foggy Cityscapes** Sakaridis et al. (2018) renders fog at three density levels ($\beta \in \{0.005, 0.01, 0.02\}$). We use the most challenging setting, $\beta$=0.02, which gives 492 evaluation images. **Clipart** contains 1,000 unlabeled images and shares the same 20 categories as PASCAL. For both Foggy Cityscapes and Clipart, we follow the PASCAL evaluation protocol.

### 4.2. Implementation Details

**Hyperparameters**: We adopt Faster R-CNN Ren et al. (2016) with a ResNet-50 backbone for all experiments. The model is optimized with SGD and momentum. We use a batch size of 2 and a learning rate of $1 \times 10^{-4}$ across all datasets. Unless otherwise stated, we set $\lambda_{st}^{cls} = \lambda_{st}^{reg} = 4.0$, $\lambda_{al} = 0.1$, $\lambda_{con} = 0.1$, and $\gamma = \frac{1}{64}$.

**Data Augmentation**: Following the semi-supervised detection setting, we apply both weak and strong augmentations. The strong augmentation includes scale jittering, solarization, brightness/contrast/sharpness jittering, translation, rotation, and RandErase, where fewer than five random patches are removed to simulate occlusion. The weak augmentation contains only random resizing and horizontal flipping, allowing the teacher model to generate more reliable pseudo-labels for supervising the student model.

**Competing Methods**: For UDA and source-free baselines, we directly report the numbers from the original papers. For TTA baselines, public TTAOD implementations remain limited, so we re-implement representative methods under the same detector and evaluation protocol. **Source Only** performs direct inference on the target domain without adaptation. **BN** Ioffe and Szegedy (2015) updates backbone batch-normalization statistics using target samples. **TENT** Wang et al. (2020) adapts batch-normalization parameters by entropy minimization. **T3A** Iwasawa and Matsuo (2021) updates class prototypes online for prediction. **SHOT** Liang et al. (2020) freezes the classifier and adapts the feature extractor with pseudo-labels and category balancing. **Self-Training** Xu et al. (2021) is adapted from semi-supervised object detection by retaining only its unsupervised component. **TTAC** Su et al. (2022) aligns source and target clusters for online adaptation. **SAR** Niu et al. (2023) improves stability through sharpness-aware entropy minimization and reliable-sample filtering. **CoTTA** Wang et al. (2022) combines entropy minimization, EMA-teacher supervision, and stochastic restoration. **RoTTA** Yuan et al. (2023) further introduces a memory queue of recent target samples for more robust online updates. **WHW** Yoo et al. (2024) uses lightweight adaptors, class-wise feature alignment, and dynamic update triggering.

### 4.3. Main Results

**COCO → COCO-C (Individual)**: Table 1 reports adaptation results for each corruption independently. Corruption causes a severe performance drop, reducing the clean-source mAP from 44.6% to 15.9% for Source Only. BN, TENT, and T3A perform poorly in this setting, indicating that updating only normalization statistics or classifier-related parameters is insufficient for corrupted object detection. In contrast, SHOT, Self-Training, TTAC, and WHW all improve over direct testing, confirming the benefit of stronger feature adaptation or pseudo-label supervision. TTAC is a particularly strong baseline, suggesting that distribution alignment remains effective under synthetic corruptions. Notably, TTAC outperforms STFAR on two digital corruptions (Pixelate: 25.4 vs. 23.0, and JPEG compression: 23.7 vs. 23.0). We attribute this to TTAC's stronger emphasis on global distribution matching, which can be advantageous

Table 1: Results on **COCO → COCO-C** under individual and continual corruptions (mAP, %).

| **Methods** | **Noise** | | | **Blur** | | | | **Weather** | | | | **Digital** | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Gauss | Shot | Impulse | Defocus | Glass | Motion | Zoom | Snow | Frost | Fog | Bright | Contrast | Elastic | Pixelate | Jpeg | |
| Clean | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 44.6 |
| Source Only | 8.2 | 10.0 | 9.1 | 12.9 | 4.7 | 9.1 | 4.9 | 19.8 | 24.0 | 38.9 | 38.4 | 22.9 | 16.5 | 6.2 | 13.2 | 15.9 |
| **Individual Adaptation** | | | | | | | | | | | | | | | | |
| BN Ioffe and Szegedy (2015) | 1.4 | 1.8 | 1.5 | 1.7 | 0.8 | 1.8 | 2.0 | 5.8 | 8.3 | 13.6 | 15.2 | 3.4 | 7.3 | 2.2 | 3.1 | 4.7 |
| TENT Wang et al. (2020) | 1.5 | 1.7 | 1.6 | 0.5 | 0.5 | 1.6 | 0.8 | 5.4 | 6.4 | 9.7 | 8.5 | 5.6 | 5.0 | 2.4 | 2.2 | 3.6 |
| T3A Iwasawa and Matsuo (2021) | 4.6 | 5.8 | 5.2 | 8.3 | 3.1 | 5.8 | 3.5 | 13.8 | 17.2 | 28.9 | 28.8 | 15.9 | 11.3 | 4.1 | 9.0 | 11.0 |
| SHOT Liang et al. (2020) | 11.0 | 13.0 | 12.1 | 14.7 | 7.2 | 11.0 | 6.4 | 22.0 | 26.7 | **41.5** | **40.9** | 26.6 | 19.7 | 9.7 | 16.4 | 18.6 |
| Self-Training Xu et al. (2021) | 13.8 | 15.3 | 15.3 | 14.5 | 9.7 | 11.9 | 6.6 | 24.8 | 26.0 | 38.4 | 37.4 | 28.5 | 20.2 | 14.5 | 18.9 | 19.7 |
| TTAC Su et al. (2022) | 14.3 | 16.2 | 15.3 | 14.2 | 11.9 | 13.2 | 7.3 | 24.0 | 26.9 | 39.0 | 38.9 | 28.3 | 26.2 | **25.4** | **23.7** | 21.7 |
| WHW Yoo et al. (2024) | 13.6 | 15.6 | 14.8 | 14.3 | 13.6 | 14.3 | **7.8** | 24.0 | 26.7 | 37.5 | 36.8 | 27.0 | **27.3** | 23.7 | 22.6 | 21.3 |
| SAR Niu et al. (2023) | 12.8 | 14.5 | 14.1 | 13.2 | 10.2 | 11.8 | 6.7 | 22.9 | 25.1 | 38.5 | 37.8 | 27.0 | 21.4 | 16.1 | 19.8 | 19.5 |
| STFAR (Ours) | **16.1** | **17.8** | **17.3** | **15.2** | **13.7** | **14.7** | 7.6 | **27.8** | **28.2** | 39.1 | 38.5 | **30.3** | 27.6 | 23.0 | 23.0 | **22.7** |
| **Continual Adaptation** | | | | | | | | | | | | | | | | |
| Self-Training Xu et al. (2021) | 9.6 | 12.5 | 12.0 | 4.0 | 2.9 | 4.8 | 3.1 | 16.2 | 23.5 | 35.1 | 34.0 | 21.8 | 16.6 | 8.2 | 12.7 | 14.5 |
| WHW Yoo et al. (2024) | 12.7 | **17.8** | **17.5** | 12.4 | 11.5 | 11.3 | 6.6 | 22.8 | 26.9 | 38.6 | 38.5 | 28.0 | 25.1 | 21.2 | 22.2 | 20.9 |
| WHW-Skip Yoo et al. (2024) | 14.4 | 17.1 | 16.0 | 13.9 | 11.7 | 12.2 | 6.3 | 22.1 | 25.5 | 37.7 | 37.1 | 25.5 | 24.1 | **23.1** | 21.1 | 20.5 |
| CoTTA Wang et al. (2022) | 11.9 | 15.8 | 15.1 | 11.2 | 10.3 | 10.8 | 5.9 | 21.4 | 24.7 | 37.2 | 36.8 | 26.1 | 23.0 | 19.6 | 20.3 | 19.3 |
| RoTTA Yuan et al. (2023) | 12.3 | 16.2 | 15.6 | 11.8 | 10.8 | 11.1 | 6.1 | 21.9 | 25.2 | 37.8 | 37.4 | 26.8 | 23.7 | 20.4 | 21.0 | 19.9 |
| TCA Ni et al. (2025) | 13.4 | 17.0 | 16.4 | 12.7 | 11.4 | 11.7 | 6.4 | 22.6 | 25.9 | 38.4 | 38.0 | 27.6 | 24.3 | 21.1 | 22.1 | 20.6 |
| STFAR + StoRestore(Ours) | **15.1** | **17.8** | 17.0 | **14.7** | **12.0** | **13.4** | **7.2** | **26.0** | **28.1** | **39.6** | **39.2** | **30.3** | **25.5** | 23.0 | **22.7** | **22.1** |

Table 2: Results on **PASCAL → PASCAL-C** with a ResNet-50 backbone (mAP, %).

| **Methods** | **Noise** | | | **Blur** | | | | **Weather** | | | | **Digital** | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Gauss | Shot | Impulse | Defocus | Glass | Motion | Zoom | Snow | Frost | Fog | Bright | Contrast | Elastic | Pixelate | Jpeg | |
| Clean | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 80.4 |
| Source Only | 11.7 | 16.4 | 14.0 | 16.7 | 13.5 | 18.8 | 25.7 | 37.7 | 44.1 | 64.1 | 69.5 | 23.8 | 43.4 | 26.0 | 38.6 | 30.9 |
| BN Ioffe and Szegedy (2015) | 4.7 | 6.8 | 5.1 | 7.4 | 4.5 | 9.8 | 13.4 | 19.1 | 22.1 | 35.3 | 39.5 | 20.6 | 17.1 | 9.1 | 10.5 | 15.0 |
| TENT Wang et al. (2020) | 3.1 | 4.0 | 3.3 | 2.6 | 2.5 | 5.3 | 4.8 | 12.8 | 13.7 | 19.0 | 19.6 | 9.9 | 11.0 | 8.8 | 4.5 | 8.3 |
| T3A Iwasawa and Matsuo (2021) | 6.1 | 8.4 | 6.5 | 11.0 | 6.4 | 10.1 | 13.8 | 16.8 | 20.6 | 32.7 | 36.9 | 12.5 | 19.7 | 13.2 | 14.8 | 15.3 |
| SHOT Liang et al. (2020) | 12.0 | 19.9 | 16.4 | 18.9 | 11.6 | 19.7 | 27.6 | 42.5 | 45.8 | 67.5 | 72.0 | 31.7 | 46.6 | 33.1 | 41.8 | 33.8 |
| Self-Training Xu et al. (2021) | 27.8 | 30.6 | 33.6 | 24.2 | 25.8 | 25.8 | 27.8 | 49.1 | 51.9 | 68.8 | 71.6 | 46.4 | 57.9 | 52.8 | 53.7 | 43.2 |
| TTAC Su et al. (2022) | 29.8 | 35.1 | 33.6 | 29.3 | 28.7 | 29.2 | 34.9 | 48.0 | 50.4 | 68.7 | **72.2** | 40.4 | 58.1 | 46.1 | 46.4 | 43.4 |
| SAR Niu et al. (2023) | 31.2 | 37.4 | 35.0 | 31.8 | 29.5 | 30.8 | 35.6 | 50.2 | 51.8 | 69.1 | 71.5 | 43.7 | 58.8 | 48.5 | 47.8 | 44.9 |
| STFAR (Ours) | **34.1** | **40.5** | **35.9** | **35.9** | **32.0** | **34.6** | **36.5** | **52.8** | **53.5** | **69.5** | 72.1 | **50.8** | **59.8** | **54.9** | **54.3** | **47.8** |

for structured digital artifacts, whereas STFAR balances self-training, GFA, and PFC to improve robustness over a broader range of corruptions. This also suggests a useful direction for strengthening GFA on structured artifacts while preserving STFAR's overall robustness. Overall, STFAR achieves the best average performance.

**COCO → COCO-C (Continual)**: We next consider continual adaptation, where the 15 corruption types are encountered sequentially. This setting is more challenging because the model must adapt to the current corruption while retaining knowledge acquired from earlier ones. To improve stability, we incorporate Stochastic Restoration Wang et al. (2022), which randomly resets a small fraction of trainable weights toward their source-model values and thus mitigates catastrophic forgetting at negligible extra cost. Self-Training degrades substantially in this regime and eventually performs worse than the source model on later corruptions. WHW and WHW-Skip are more stable, but STFAR still delivers the best average result. We further compare with CoTTA, RoTTA, and TCA, all built on the same self-training detector. CoTTA and RoTTA perform below STFAR, suggesting that classification-oriented continual TTA strategies alone are less effective at handling the localization errors and instance-level ambiguity accumulated in continual object detection. TCA Ni et al. (2025), a recent continual TTA method that preserves inter-class topology during online adaptation, achieves 20.6% average mAP, which is higher than CoTTA and RoTTA but still below STFAR + StoRestore (22.1%). This indicates that while topology-preserving regularization can improve continual robustness to some extent, it remains less effective than a detection-specific design that explicitly handles pseudo-box noise, localization drift, and foreground-background confusion.

**PASCAL → PASCAL-C**: Table 2 shows that corruption also causes a large accuracy drop on PASCAL-C, where Source Only falls from 80.4% on the clean set to 30.9% on average. Self-Training and TTAC already provide clear gains over direct testing, confirming that online adaptation is effective in this setting. The largest improvements appear under more severe cor-

ruptions, where the gap between source and target distributions becomes larger. STFAR achieves the highest average performance and performs best on most corruption types, showing that the combination of self-training, GFA, and PFC transfers well beyond COCO-C.

**Cityscapes → Foggy Cityscapes**: We further evaluate STFAR under a real-world weather shift. As shown in Table 3, Source Only reaches only 25.2% mAP, which highlights the difficulty of transferring a clear-weather detector directly to foggy scenes. UDA methods still obtain the strongest results because they can access both source and target data during training; for example, MeGA-CDA reaches 41.8% mAP. Among source-free baselines, IRG achieves 37.1%. STFAR reaches 40.7% mAP without using source images or offline target-domain adaptation, outperforming most source-free baselines and approaching the performance of strong UDA methods. Among TTA baselines, STFAR also outperforms TENT (23.0%), TTAC (35.6%), and SAR (33.4%), demonstrating its effectiveness under real-world domain shifts.

**Cityscapes → Rainy Cityscapes → Foggy Cityscapes (Continual)**: We also evaluate a real-world continual shift sequence from clear weather to rain and then fog. Table 4 shows that STFAR improves over the self-training baseline on Foggy Cityscapes by 1.8 points and preserves Cityscapes performance better by 6.0 points. Although self-training is slightly better at the intermediate Rainy Cityscapes stage, it degrades more severely after the shift to fog and forgets more of the original source-domain knowledge. In contrast, when STFAR is equipped with the Stochastic Restoration strategy from CoTTA Wang et al. (2022), parameter drift is better controlled by stochastically restoring a fraction of weights toward the source model, leading to more stable adaptation across the sequential domain shifts.

**PASCAL → Clipart**: Finally, we evaluate cross-style adaptation from photographs to clipart images. As shown in Table 5, Source Only achieves only 27.8% mAP, confirming the large appearance gap between the two domains. UDA and source-free baselines improve on this result, with IRG reaching 31.5% mAP among source-free methods. STFAR further boosts performance to 37.4%, outperforming all compared baselines and yielding particularly strong gains on categories such as person and horse.

### *4.4. Qualitative Results*

We visualize object detection results on corrupted COCO images in Fig. 3. Each column group corresponds to one corruption type, and each row shows the output of Source Only, Self-Training, or STFAR. The left image presents detection results with green boxes for true positives (TP), red boxes for false positives (FP), and blue boxes for false negatives (FN, i.e., missed ground-truth objects), while the right image shows the corresponding Grad-CAM heatmap. The three scenes span diverse categories and scales: medium dogs and large persons (impulse noise), a large train with tiny traffic lights (snow), and a multi-scale elephant herd (JPEG compression). Source Only struggles most with small and medium instances: under snow, the tiny traffic lights are almost entirely missed (blue FN), and under impulse noise, the medium-sized dogs are poorly localized; its heatmaps scatter over backgrounds rather than objects. Self-Training recovers some of these missed detections but overshoots on large, salient objects: in the snow scene the train and surrounding region attract spurious FP boxes, and in the elephant scene repeated FP detections appear around already-detected individuals. STFAR consistently detects objects across all scales with fewer FP and FN than the baselines: the small traffic lights under snow are largely recovered, medium dogs under impulse noise are better localized, and the varying-size elephants under JPEG compression are more cleanly separated. While STFAR occasionally still misses some instances (e.g., the smallest traffic lights), its Grad-CAM maps concentrate on actual object regions, reflecting that GFA and PFC regularization produce more discriminative, scale-aware feature representations.

### *4.5. Ablation Studies*

We conduct an ablation study on PASCAL-C to assess the effectiveness of each individual component. As shown in Table 6, directly testing the source model on the target domain without any adaptation yields a low mAP of 11.7%. Introducing self-training alone significantly boosts performance to 27.8%, highlighting its effectiveness in adapting the source model to the target distribution. Applying global feature alignment independently also results in a notable improvement (26.9%), though slightly lower than self-training. When self-training and global feature alignment are combined, the model improves further to 32.6% mAP, demonstrating their complementary effects, with global alignment acting as a regularizer for the noisy pseudo-labels generated by self-training. Finally, incorporating Prototype-guided Foreground Contrast (PFC) leads to the best result, achieving a mAP of 34.1%. This confirms that PFC further enhances test-time adaptation by improving foreground feature discriminability, and that the combination of self-training, GFA, and PFC is highly effective for adaptive object detection under domain shift.

To provide a more intuitive view of the proposed strategy, we visualize the feature distributions using t-SNE. As shown in Fig. 4, we present feature distributions from different FPN layers under several conditions, using zoom blur and Gaussian noise as representative corruptions. Here, **Original** denotes the feature distribution of the source model on clean images, **Zoom or Gauss** denotes the distribution of the source model on corrupted images, and **Adapt** denotes the distribution of the adapted model on corrupted images. A clear discrepancy can be observed between the clean and corrupted feature distributions at every FPN layer. In contrast, feature alignment pulls corrupted-image features closer to the source-domain distribution, indicating its effectiveness in mitigating domain shift. We further visualize instance-level feature distributions under elastic corruption in Fig. 5. Corruption makes the decision boundaries less distinct and increases ambiguity among visually similar categories. The proposed PFC alleviates this issue by producing more separable and discriminative feature representations. In addition, Fig. 6 compares recall and precision against

Table 3: Results on **Cityscapes → Foggy Cityscapes** (mAP, %). S: Source Only; UDA: Unsupervised Domain Adaptation; SFDA: Source-Free Domain Adaptation; TTA: Test-Time Adaptation.

| Type | Method | prsn | rider | car | truck | bus | train | mcycle | bicycle | mAP |
|---|---|---|---|---|---|---|---|---|---|---|
| S | Source Only | 29.3 | 34.1 | 35.8 | 15.4 | 26.0 | 9.1 | 22.4 | 29.7 | 25.2 |
| UDA | DA FasterChen et al. (2018) | 25.0 | 31.0 | 40.5 | 22.1 | 35.3 | 20.2 | 20.0 | 27.1 | 27.6 |
| | SWDASaito et al. (2019) | 29.9 | 42.3 | 43.5 | 24.5 | 36.2 | 32.6 | 30.0 | 35.3 | 34.3 |
| | Progressive DA Hsu et al. (2020) | 36.0 | 45.5 | 54.4 | 23.2 | 45.7 | 25.8 | 29.1 | 35.9 | 36.9 |
| | Categorical DA Xu et al. (2020) | 32.9 | 43.8 | 49.2 | 27.2 | 45.1 | 36.4 | 30.3 | 34.6 | 37.4 |
| | MeGA CDA Hsu et al. (2020) | 37.7 | 49.0 | 52.4 | 25.4 | 49.2 | 46.9 | 34.5 | 39.0 | 41.8 |
| | Unbiased DA Deng et al. (2021) | 33.8 | 47.3 | 49.8 | 30.0 | 48.2 | 42.1 | 33.0 | 37.3 | 40.4 |
| SFDA | SFOD-MosaicLi et al. (2021) | 21.7 | 44.0 | 40.4 | 32.2 | 11.8 | 25.3 | 34.5 | 34.3 | 30.6 |
| | HCLHuang et al. (2021) | 26.9 | 46.0 | 41.3 | 33.0 | 25.0 | 28.1 | 35.9 | 34.6 | 34.6 |
| | LODSLi et al. (2022) | 34.0 | 45.7 | 48.8 | 27.3 | 39.7 | 19.6 | 33.2 | 37.8 | 35.8 |
| | Mean-TeacherXu et al. (2021) | 33.9 | 43.0 | 45.0 | 29.2 | 27.1 | 25.6 | 38.2 | 34.3 | 37.1 |
| | IRGVS et al. (2023) | 37.4 | 46.9 | 51.9 | 24.4 | 39.6 | 25.2 | 41.5 | 41.6 | 37.1 |
| TTA | TENT Wang et al. (2020) | 25.8 | 31.2 | 33.4 | 12.1 | 28.4 | 5.3 | 20.1 | 27.8 | 23.0 |
| | TTAC Su et al. (2022) | 38.5 | 47.2 | 53.1 | 22.4 | 44.3 | 13.8 | 27.3 | 37.8 | 35.6 |
| | SAR Niu et al. (2023) | 36.2 | 44.8 | 50.3 | 20.7 | 41.2 | 12.4 | 25.8 | 35.6 | 33.4 |
| | STFAR (Ours) | 43.2 | 51.0 | 58.4 | 28.9 | 50.6 | 21.2 | 30.7 | 41.7 | 40.7 |

Table 4: Results on Cityscapes → Rainy Cityscapes → Foggy Cityscapes under continual adaptation (mAP, %).

| Method | Rainy Cityscapes | Foggy Cityscapes | Source Domain |
|---|---|---|---|
| Source Only | 35.1 | 31.7 | 65.6 |
| Self-Training | 39.6 | 35.4 | 54.4 |
| STFAR + StoRestore | 39.4 | 37.2 | 60.4 |

self-training, showing that the proposed method consistently outperforms the self-training baseline on both metrics.

**Alternative Weights Update**: By default, STFAR updates only the backbone weights during TTA, since this allows the RPN and R-CNN heads to be reused and these components are less likely to be directly affected by target-domain corruptions. In this section, we examine alternative subsets of weights to update during TTA. As shown in Table 7, when BatchNorm statistics are updated, mAP drops significantly. When BN statistics are frozen and only the affine parameters are updated, performance improves over the baseline. When all model weights, including RPN and R-CNN, are updated, the performance is still inferior to STFAR, which only updates the backbone weights. Overall, updating only the backbone weights during TTA is more effective than the alternative update strategies considered here.

**Feature Layers for Alignment:** To identify the most suitable global features for object detection, we evaluate feature alignment at different FPN layers (see Table 8). Specifically, in Faster R-CNN with a Feature Pyramid Network (FPN), we align features from different pyramid levels. Experimental results show that aligning features from different layers consistently outperforms the unaligned baseline, indicating that multi-layer features can all be effectively utilized for object detection. Additionally, aligning higher-level FPN features generally yields better performance, suggesting that higher-level semantic features are more amenable to distribution modeling. Moreover, aligning features from all layers simultaneously achieves the best results, as it imposes richer constraints that improve model performance.

**Contrastive Head Dimension:** Table 9 shows performance under different contrastive-head feature lengths. The best result is achieved with a length of 128, yielding 34.1% mAP, and we therefore use this setting by default. Other settings lead to slightly lower but comparable results, indicating that moderate feature dimensions are more effective for contrastive learning.

**Stability of the TTA Algorithm:** We also investigate the stability of the STFAR algorithm by running it multiple times with different random seeds. Table 10 presents the mean Average Precision (mAP) over five different seeds evaluated on the Gaussian Noise subset of PASCAL-C. All seeds produced similar results, with the mAP values ranging from 33.7% to 34.4%. These results suggest that the model is relatively stable across random seeds, as performance varies only slightly between runs.

## 5. Conclusion

This work addresses the challenge of adapting a source-domain detector to target data corrupted by natural noise when the target distribution is unknown before inference. By combining self-training with feature-distribution alignment regularization, the detector can be updated effectively during test-time adaptation. To support systematic evaluation, we transform three standard object detection datasets into corrupted target-domain benchmarks for TTAOD. Comprehensive experiments show that the proposed method consistently outperforms existing test-time adaptation baselines, demonstrating strong robustness and effectiveness. Despite these advantages, the method introduces additional computational overhead and does not yet

Table 5: Results on **PASCAL** → **Clipart** (mAP, %). S: Source Only; UDA: Unsupervised Domain Adaptation; SFDA: Source-Free Domain Adaptation; TTA: Test-Time Adaptation.

| Type | Method | aero | bcycle | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | bike | prsn | plnt | sheep | sofa | train | tv | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S | Source Only | 35.6 | 52.5 | 24.3 | 23.0 | 20.0 | 43.9 | 32.8 | 10.7 | 30.6 | 11.7 | 13.8 | 6.0 | 36.8 | 45.9 | 48.7 | 41.9 | 16.5 | 7.3 | 22.9 | 32.0 | 27.8 |
| UDA | DA FasterChen et al. (2018) | 15.0 | 34.6 | 12.4 | 11.9 | 19.8 | 21.1 | 23.3 | 3.1 | 22.1 | 26.3 | 10.6 | 10.0 | 19.6 | 39.4 | 34.6 | 29.3 | 1.0 | 17.1 | 19.7 | 24.8 | 19.8 |
| | ADDAInoue et al. (2018) | 20.1 | 50.2 | 20.5 | 23.6 | 11.4 | 40.5 | 34.9 | 2.3 | 39.7 | 22.3 | 27.1 | 10.4 | 31.7 | 53.6 | 46.6 | 32.1 | 18.0 | 21.1 | 23.6 | 18.3 | 27.4 |
| | BDC FasterSaito et al. (2019) | 20.2 | 46.4 | 20.4 | 19.3 | 18.7 | 41.3 | 26.5 | 6.4 | 33.2 | 11.7 | 26.0 | 11.7 | 36.6 | 41.5 | 37.7 | 44.5 | 10.6 | 20.4 | 33.3 | 15.5 | 25.6 |
| SFDA | PL Khodabandeh et al. (2019) | 18.3 | 48.4 | 19.2 | 22.4 | 12.8 | 38.9 | 36.1 | 5.2 | 36.9 | 24.8 | 29.3 | 9.1 | 34.6 | 58.6 | 43.1 | 34.3 | 9.1 | 14.4 | 26.9 | 19.8 | 28.2 |
| | SFOD Li et al. (2021) | 20.1 | 51.5 | 26.8 | 23.0 | 24.8 | 64.1 | 37.6 | 10.3 | 36.3 | 20.0 | 18.7 | 13.5 | 26.5 | 49.1 | 37.1 | 32.1 | 10.1 | 17.6 | 42.6 | 30.0 | 29.5 |
| | Mean-Teacher Xu et al. (2021) | 22.3 | 42.3 | 23.8 | 21.7 | 23.5 | 60.7 | 33.2 | 9.1 | 24.7 | 16.7 | 12.2 | 13.1 | 26.8 | 73.6 | 43.9 | 34.5 | 9.1 | 24.3 | 37.9 | 42.2 | 29.1 |
| | IRG VS et al. (2023) | 20.3 | 47.3 | 27.3 | 19.7 | 30.5 | 54.2 | 36.2 | 10.3 | 35.1 | 20.6 | 20.2 | 12.3 | 28.7 | 53.1 | 47.5 | 42.4 | 9.1 | 21.1 | 42.3 | 50.3 | 31.5 |
| TTA | STFAR (Ours) | 13.6 | 58.2 | 29.4 | 29.3 | 40.8 | 54.7 | 39.4 | 10.9 | 34.8 | 8.1 | 36.3 | 25.3 | 42.9 | 77.0 | 61.0 | 41.7 | 23.3 | 26.3 | 44.9 | 50.4 | 37.4 |

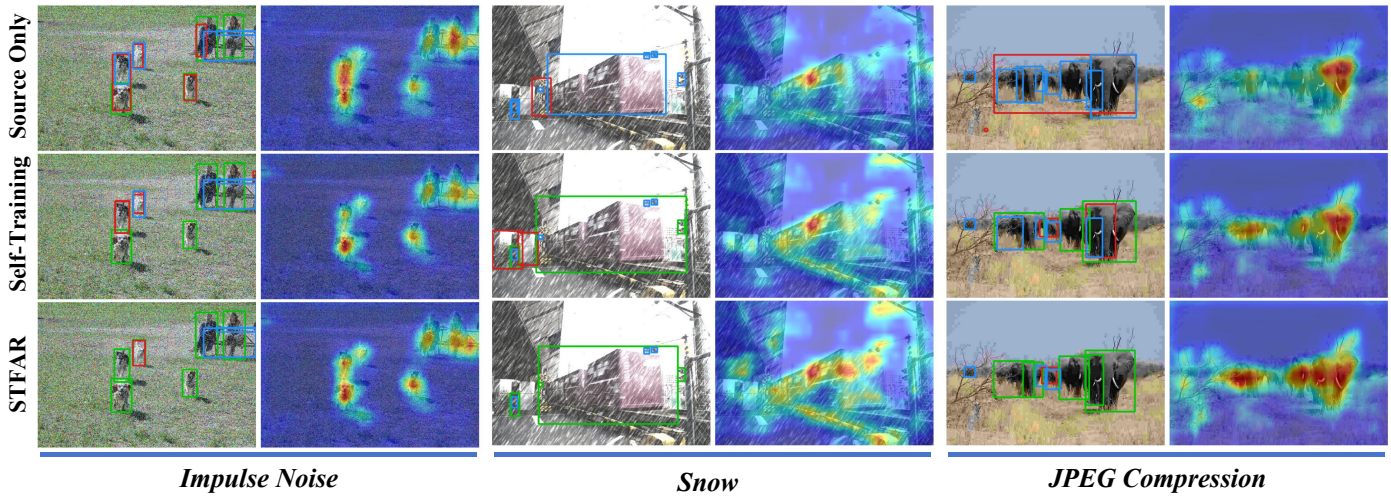


Figure 3: Qualitative results on **COCO-C** under three corruption types: impulse noise, snow, and JPEG compression. For each row, the left image shows detection results with green boxes for true positives (TP), red for false positives (FP), and blue for false negatives (FN, missed ground-truth objects), while the right image shows the corresponding Grad-CAM activation heatmap.

Table 6: Ablation on PASCAL-C using the Gaussian Noise subset of the validation set (mAP, %).

| Self Training | Global Feature Alignment | Prototype-guided Foreground Contrast | **mAP** |
|---|---|---|---|
| – | – | – | 11.7 |
| ✓ | – | – | 27.8 |
| – | ✓ | – | 26.9 |
| ✓ | ✓ | – | 32.6 |
| ✓ | ✓ | ✓ | **34.1** |

Table 7: Effect of update parameter choices on **PASCAL-C** using Gaussian Noise (mAP, %).

| Methods | mAP |
|---|---|
| Source Only | 11.7 |
| Update BatchNorm (BN) statistics | 4.7 |
| Update BN affine parameters (with updated statistics) | 6.3 |
| Update BN affine parameters (with frozen statistics) | 27.6 |
| Update all network weights except BN layers | 30.2 |
| Update backbone weights except BN layers | **34.1** |

leverage external semantic priors, which may limit efficiency in real-time applications. Future work will focus on reducing computational and memory costs and on incorporating knowledge from foundation or vision-language models to provide richer semantic guidance in real-world scenarios.

## References


Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K., 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning, in: International Joint Conference on Neural Networks.

Cai, J., Li, H., Tan, M., He, B., Lv, W., Li, H., 2025. Cross-modal generalizable medical image segmentation with dual-domain deformable transformer and multitask adaptation. Expert Systems with Applications 277, 127249. doi:10.1016/j.eswa.2025.127249.

Chen, C., Zheng, Z., Ding, X., Huang, Y., Dou, Q., 2020. Harmonizing transferability and discriminability for adapting object detectors, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

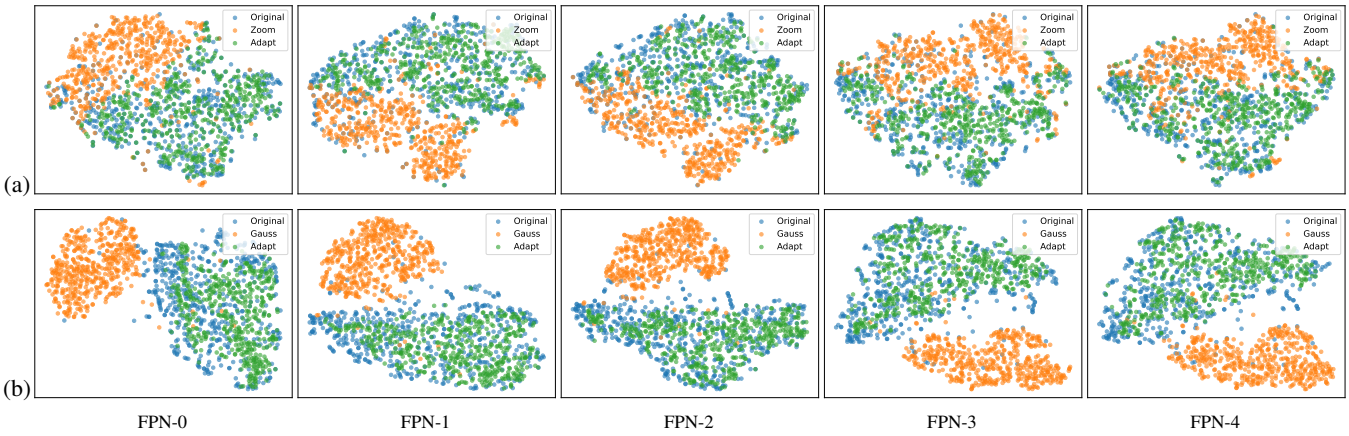


Figure 4: t-SNE visualizations on two corruptions: (a) Zoom blur and (b) Gaussian noise. The plots show the feature distribution of clean images in the source-domain model (**Original**), corrupted images in the source-domain model (**Zoom or Gauss**), and corrupted images in the adapted model (**Adapt**).

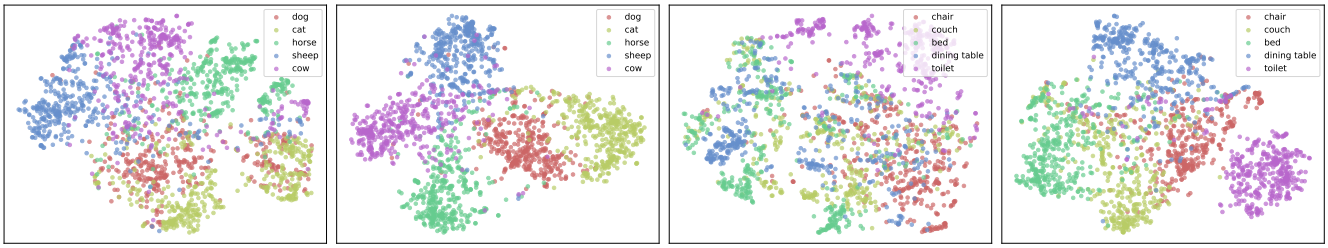


Figure 5: t-SNE visualizations on elastic corruption in **COCO-C**.

Table 8: Effect of global feature alignment layers on PASCAL-C using the Gaussian-corrupted validation subset (mAP, %).

| Method | F0 | F1 | F2 | F3 | F4 | mAP |
|---|---|---|---|---|---|---|
| Source Only | | | | | | 11.7 |
| Self-Training | | | | | | 27.8 |
| Align @ F0 | ✓ | | | | | 30.3 |
| Align @ F1 | | ✓ | | | | 31.8 |
| Align @ F2 | | | ✓ | | | 32.6 |
| Align @ F3 | | | | ✓ | | 31.6 |
| Align @ F4 | | | | | ✓ | 32.0 |
| Align @ All | ✓ | ✓ | ✓ | ✓ | ✓ | **34.1** |

Table 9: Effect of contrastive head feature length on **PASCAL-C** using Gaussian Noise (mAP, %).

| Length | 64 | 128 | 256 | 512 | 1024 |
|---|---|---|---|---|---|
| mAP | 33.4 | **34.1** | 33.7 | 33.6 | 33.6 |

Table 10: Random-seed sensitivity on PASCAL-C using Gaussian Noise (mAP, %).

| Clean | Source | S0 | S1 | S42 | S789 | S2000 | Avg±Std |
|---|---|---|---|---|---|---|---|
| 80.4 | 11.7 | 33.9 | 34.4 | 34.1 | 33.7 | 34.1 | 34.0±0.3 |

Chen, Y., Li, W., Sakaridis, C., Dai, D., Gool, L.V., 2018. Domain adaptive faster r-cnn for object detection in the wild, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Deng, J., Li, W., Chen, Y., Duan, L., 2021. Unbiased mean teacher for cross-domain object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4091–4101.

Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2015. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision .

Fan, D., Liu, M., Shao, Y., Yang, L., Liu, Y., Zhang, Y., Ren, Y., Wang, Z., 2025. Domain-specific large language model for maintenance decision-making on wind farms by labeled-data-supervised fine-tuning. Engineering doi:10.1016/j.eng.2025.12.019.

Fortes, V., Rezende, P.M.B., Zhou, B., Suh, S., Lukowicz,

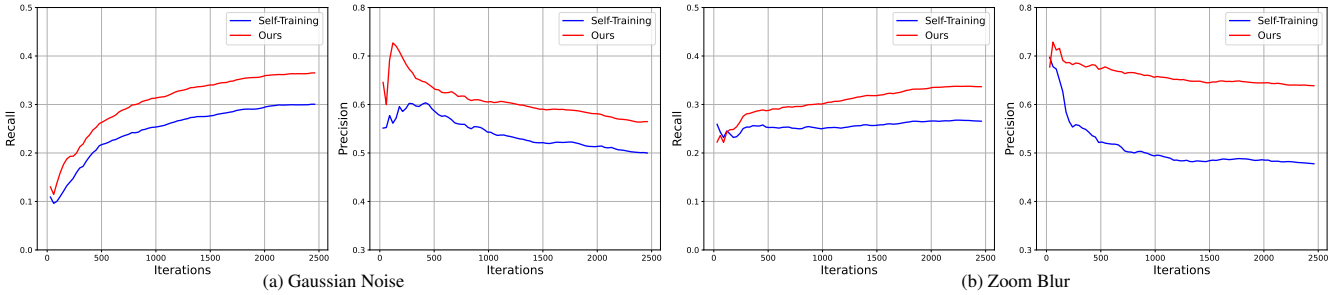


Figure 6: Comparison of recall and precision on **PASCAL-C**: (a) Gaussian noise and (b) Zoom blur.

P., 2026. Coa-har: Exploring contrastive online test-time adaptation for wearable sensor-based human activity recognition using sensor data augmentation. Expert Systems with Applications 297, 129288. doi:10.1016/j.eswa.2025.129288.

Gong, T., Jeong, J., Kim, T., Kim, Y., Shin, J., Lee, S.J., 2022. Note: Robust continual test-time adaptation against temporal correlation, in: Advances in Neural Information Processing Systems.

Hou, Y., Wang, Y., Xia, X., Tian, Y., Li, Z., Quek, T.Q.S., 2026. Toward secure SAR image generation via federated angle-aware generative diffusion framework. IEEE Internet of Things Journal 13, 2713–2730.

Hsu, H.K., Yao, C.H., Tsai, Y.H., Hung, W.C., Tseng, H.Y., Singh, M., Yang, M.H., 2020. Progressive domain adaptation for object detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV).

Huang, J., Guan, D., Xiao, A., Lu, S., 2021. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data, in: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 3635–3649.

Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K., 2018. Cross-domain weakly-supervised object detection through progressive domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5001–5009.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, pmlr. pp. 448–456.

Iwasawa, Y., Matsuo, Y., 2021. Test-time classifier adjustment module for model-agnostic domain generalization. Advances in Neural Information Processing Systems 34, 2427–2440.

Khodabandeh, M., Vahdat, A., Ranjbar, M., Macready, W.G., 2019. A robust learning approach to domain adaptive object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision.

Li, S., Ye, M., Zhu, X., Zhou, L., Xiong, L., 2022. Source-free object detection by learning to overlook domain style, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Li, X., Chen, W., Xie, D., Yang, S., Yuan, P., Pu, S., Zhuang, Y., 2021. A free lunch for unsupervised domain adaptive object detection without source data, in: Proceedings of the AAAI Conference on Artificial Intelligence.

Li, Y., Long, S., Wang, S.S., Zhao, X., Li, Y., 2025. Prompt-induced prototype alignment for few-shot unsupervised domain adaptation. Expert Systems with Applications 269, 126400. doi:10.1016/j.eswa.2025.126400.

Liang, J., Hu, D., Feng, J., 2020. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation, in: International Conference on Machine Learning.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context, in: European Conference on Computer Vision.

Mushtaq, H., Deng, X., Alizadehsani, R., Iqbal, M.S., Khan, T., Abbasi, A.A., 2025. Sc3d: Semantic-guided and class-adaptive cross-domain fusion for 3d object detection in autonomous vehicles. Expert Systems with Applications 268, 126359. doi:10.1016/j.eswa.2024.126359.

Ni, C., Lyu, F., Tan, J., Hu, F., Yao, R., Zhou, T., 2025. Maintaining consistent inter-class topology in continual test-time adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15319–15328.

Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P., Tan, M., 2023. Towards stable test-time adaptation in dynamic wild world, in: The Eleventh International Conference on Learning Representations. URL: https://openreview.net/forum?id=g2YraF75Tj.

van den Oord, A., Li, Y., Vinyals, O., 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 .

Rajendran, G.B., Srinivasan, G.T., Niruban, R., 2025. Adaptive yolov6 with spatial transformer networks for accurate object detection and multi-angle classification in remote sensing images. Expert Systems with Applications 282, 127796. doi:10.1016/j.eswa.2025.127796.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Saito, K., Ushiku, Y., Harada, T., Saenko, K., 2019. Strong-weak distribution alignment for adaptive object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Sakaridis, C., Dai, D., Van Gool, L., 2018. Semantic foggy scene understanding with synthetic data. International Journal of Computer Vision .

Sindagi, V.A., Oza, P., Yasarla, R., Patel, V.M., 2020. Prior-based domain adaptive object detection for hazy and rainy conditions, in: Computer Vision – ECCV 2020, Springer. pp. 763–780.

Su, Y., Xu, X., Jia, K., 2022. Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering, in: Advances in Neural Information Processing Systems.

Sun, B., Li, B., Cai, S., Yuan, Y., Zhang, C., 2021. Fsce: Few-shot object detection via contrastive proposal encoding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7352–7362.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., Hardt, M., 2020. Test-time training with self-supervision for generalization under distribution shifts, in: International Conference on Machine Learning.

Tarvainen, A., Valpola, H., 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Advances in Neural Information Processing Systems .

VS, V., Oza, P., Patel, V.M., 2023. Instance relation graph guided source-free domain adaptive object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3520–3530.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T., 2020. Tent: Fully test-time adaptation by entropy minimization, in: International Conference on Learning Representations.

Wang, Q., Fink, O., Van Gool, L., Dai, D., 2022. Continual test-time domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7201–7211.

Wen, Z., Liu, J., Tang, J., Zhao, H., Jiang, L., Wang, Q., 2025a. A robust defect detection method for vision-based measurement system via wavelet-guided spatial-frequency prior network. IEEE Transactions on Instrumentation and Measurement 74, 1–11. doi:10.1109/TIM.2025.3643046.

Wen, Z., Liu, J., Zhang, H., Zuo, F., 2025b. Exploring fine-grained visual-text feature alignment with prompt tuning for domain-adaptive object detection. IEEE Transactions on Cybernetics 55, 3220–3233. doi:10.1109/TCYB.2025.3567126.

Wen, Z., Liu, J., Zhao, H., Wang, Q., 2025c. A triple semantic-aware knowledge distillation network for industrial defect detection. Computers in Industry 166, 104252. doi:10.1016/j.compind.2025.104252.

Xu, C.D., Zhao, X.R., Jin, X., Wei, X.S., 2020. Exploring categorical regularization for domain adaptive object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., Bai, X., Liu, Z., 2021. End-to-end semi-supervised object detection with soft teacher, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

Xu, N., Wu, D., Tian, Y., Quek, T.Q.S., 2026. FedC-DAC: A federated clustering with dynamic aggregation and calibration method for SAR image target recognition. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 19, 3726–3745. doi:10.1109/JSTARS.2025.3649421.

Xu, Y., Sun, Y., Yang, Z., Miao, J., Yang, Y., 2022. H2fa r-cnn: Holistic and hierarchical feature alignment for cross-domain weakly supervised object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 14329–14339.

Yang, H., Liu, Z., Cui, H., Ma, N., Wang, H., Zhang, C., Song, Y., 2025. An electrified railway catenary component anomaly detection frame based on invariant normal region prototype with segment anything model. IEEE Transactions on Transportation Electrification , 1–1doi:10.1109/TTE.2025.3628607.

Yoo, J., Lee, D., Chung, I., Kim, D., Kwak, N., 2024. What how and when should object detectors update in continually changing test domains?, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 23354–23363.

Yuan, L., Xie, B., Li, S., 2023. Robust test-time adaptation in dynamic scenarios, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15499–15508.

Zhao, X., Wang, X., 2025. Unsupervised domain adaptation based fracture segmentation method for core ct images. Expert Systems with Applications 264, 125857. doi:10.1016/j.eswa.2024.125857.

Zhao, Z., Guo, Y., Ye, J., 2020. Bi-dimensional feature alignment for cross-domain object detection, in: Computer Vision – ECCV 2020 Workshops, pp. 671–686.